# 写在前面

本篇介绍逆向技能的学习。学习目标是完毕之后，知道对于一个exe，如何把它拆开，知道它里面是什么结构，以及其中代码的含义。本篇依旧是学习指导的性质，只是给出相关的知识去哪里学、用什么关键字搜索。

# 准备知识

开始之前，我假设你已经拥有了如下技能或经验：

* 熟练掌握x86汇编指令。
* 使用Win32 API编写过窗口程序，熟练使用MSDN。
* 清楚调试器的用法。
* 熟悉PE文件格式。

然后弄清楚下面两个关键字的含义：

* 调用约定。
* 字符编码。

# OllyDbg

这是一个调试器，目前有很多个mod，这里推荐使用Immunity Debugger，因为它支持Python脚本。不过学习的时候无所谓，原版就好。可能你在学习Win32汇编的时候已经用过了，这里还是给出CCDebugger的一份[教程](http://bbs.pediy.com/showpost.php?p=514664)。如果不能下载附件的话，我在[这里](https://drive.google.com/file/d/0BwkiwidxAjbmNElpUFFZWV90LUU/edit?usp=sharing)打包了一下（解压密码OllyDbg）。

教程学完之后可以从头看下OllyDbg的帮助文档，刚刚的教程链接里面就有。搞清楚各种快捷键的使用，这在调试中很有帮助。

然后来个实战演练吧，把[这个软件](https://drive.google.com/file/d/0BwkiwidxAjbmYklfb1A4cDc5dE0/edit?usp=sharing)注册一下。不允许暴力破解。如果认真学完了CCDebugger的教程的话，应该不难。

最后进行一点进阶学习，把[这个教程](http://bbs.pediy.com/showthread.php?t=31840)的第一课（3L）和第三课（4L）看完。

学习个差不多之后，以后再写C程序，可以都直接用od调试，并看看之前用VC的调试器做到的事情如何用OD来做到。顺便OD支持源码级调试，不会的话自己搜一下。

# 脱壳

这个词一般都念“tuoke”，具体该念什么就不理会了。

加壳是保护代码不被他人逆向分析的最常用手段，因此脱壳也就成为了通常破解者首先需要掌握的技术。不过可能是考虑到预算、执行效率等因素，游戏程序加壳的不算多，因此我们这里的学习也只是浅尝辄止（好吧，其实是因为我自己很不擅长这个……）。

[这里](http://bbs.pediy.com/showthread.php?t=20366)是一篇比较系统的脱壳教学。学完之后兴许你可以自己手脱一个穿山甲之类的。但是现实中游戏加的壳一般都是商业壳，SecuROM、StarForce、Themida等等，脱这些壳需要丰富的经验和大量的时间与精力，……所以就不再深入了。

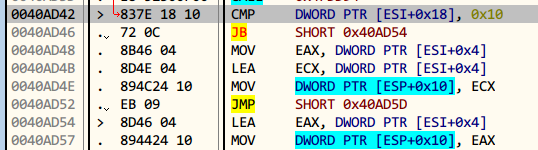
不过给一个思路：从简单的入手。比如Themida，你可以先自己用一个很早期的版本，使用最简单的加密选项，处理一个自己写的小程序，然后尝试脱掉它。之后再换用新一点的版本，重点寻找新版本有哪些改动。这样一步一步前进，会比上来直接脱一个商业版成品要容易的多。

最后，可以自己尝试写一个加密壳，目标是能在OD不带反反调试插件的情况下，把它down掉。不过自己完整的写一个壳工程量不会小，可以在一个简单的开源壳的基础上进行修改以达到目的。

# 分析能力

逆向过程中，最关键的一点就是快速的理解一系列汇编代码所代表的逻辑含义。换句话说，也就是知道什么样的编译器会把什么样的C/C++或其他源代码，编译成什么样的汇编代码。事实上我们所接触的exe绝大部分都是用微软的VC编译器编译的，不过就算是VC，不同版本生成的代码可能也会有不小的区别，这些都是需要靠经验积累的。

而除了编译器区别，还有一些代码和库有关，比如当你看到形如下面这样的代码时



首先应该理解到这可能是一个与MFC的CString类似的字符串对象。还有如果程序把zlib库编译到了exe中，则可以通过zlib的版本号字符串来简单的识别uncompress函数等等。

推荐在自己分析过100kb以上的代码之后看一下《黑客反汇编揭秘》这本书，这本书出过两版，第一版的内容相对基础一些，有一章专门讲在汇编代码中识别各种C语言结构；第二版的内容则更侧重于代码保护和攻击方面的内容，有时间的话推荐都看一看。

# 结束语

有关逆向方面的指引到这里就结束了，如果能把上面的内容全都融会贯通，并自己真正分析过几百kb的代码，那么汉化个一般的游戏什么的已经是手到擒来了。

我会在下一篇里面讲一下汉化一个游戏的具体步骤。